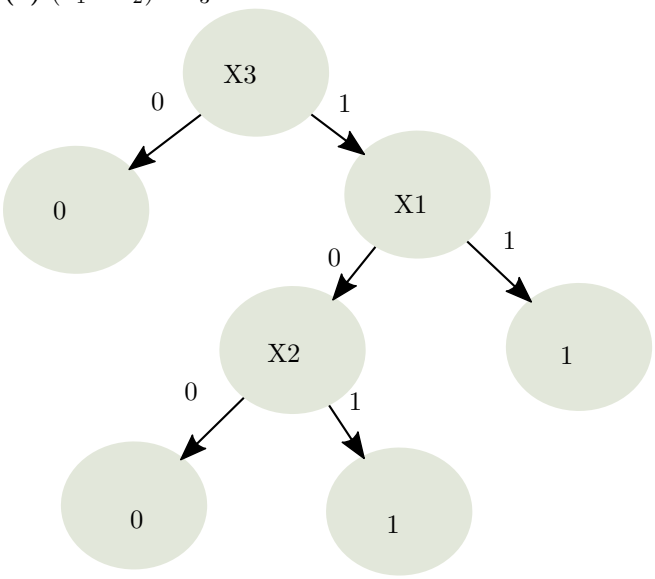
CS6210 - Homework/Assignment-1

Arnabd Das(u1014840)

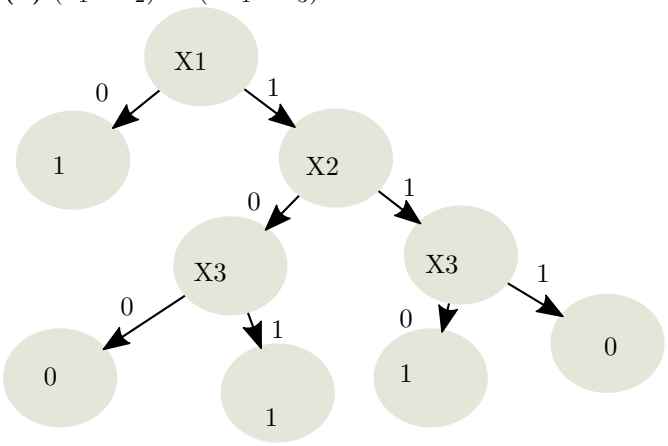
September 12, 2016

Question-1.1: DECISION TREES

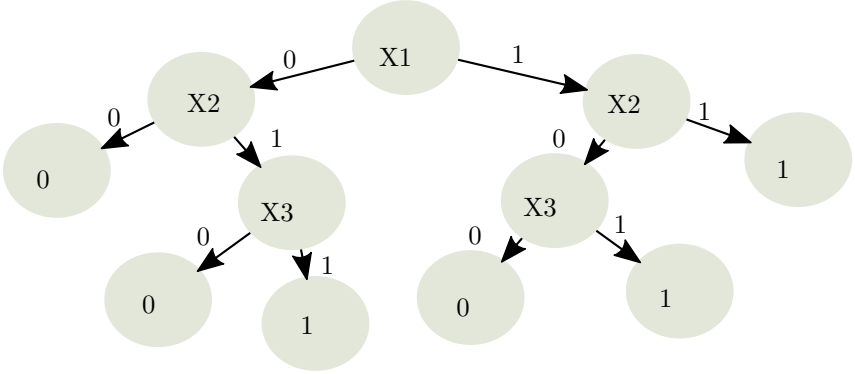
(a) $(x_1 \vee x_2) \wedge x_3$



(b) $(x_1 \wedge x_2) xor (\neg x_1 \vee x_3)$



(c) 2-of-3 function: atleast 2 of x_1, x_2, x_3 should be true for output to be true



Question-1.2: ID3: PokeMonGo

(a) Possible function to map these four features to a boolean decision = $2^{2^4} = 65536$

(b) In the Given example set, S, there are 8 yes and 8 no, out of the total 16 examples in the set.

$$\text{Entrophy(S)} = -P_{yes}\log_2 P_{yes} - P_{no}\log_2 P_{no} = -\frac{8}{16}\log_2(\frac{8}{16}) - \frac{8}{16}\log_2(\frac{8}{16}) = 1 \text{ (Answer)}$$

(c) Calculating the information gain of the 4 attributes
SELECT ATTRIBUTE : Berry ; Values Berry Takes = $< Yes, No >$; E(S) = Entrophy(S)

$\sum_{v \in values(Berry)} \left|\frac{S_v}{S}\right| E(S) = \left|\frac{S_{yes}}{S}\right| E(S_{yes}) + \left|\frac{S_{no}}{S}\right| E(S_{no})$

$= \frac{7}{16}\Big(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}\Big) + \frac{9}{16}\Big(-\frac{2}{9}\log_2\frac{2}{9} - \frac{7}{9}\log_2\frac{7}{9}\Big) = 0.6887$

$Information\_Gain(S, Berry)$ = E(S) - 0.6887 = $1 - 0.6887 = 0.3113$ (Answer)

SELECT ATTRIBUTE : Ball ; Values Ball Takes = $< Poke, Great, Ultra >$

$\left|\frac{S_{poke}}{S}\right| E(S_{poke}) + \left|\frac{S_{Great}}{S}\right| E(S_{Ultra}) + \left|\frac{S_{Ultra}}{S}\right| E(S_{Ultra})$

$= \frac{6}{16}\Big(-\frac{1}{6}\log_2\frac{1}{6} - \frac{5}{6}\log_2\frac{5}{6}\Big) + \frac{7}{16}\Big(-\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}\Big) + \frac{3}{16}\Big(-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\Big) = 0.6748$

$Information\_Gain(S, Ball)$ = E(S) - 0.6748 = $1 - 0.6748 = 0.3252$ (Answer)

SELECT ATTRIBUTE : Color ; Values Color Takes = $< Green, Yellow, Red >$

$\left|\frac{S_{Green}}{S}\right| E(S_{Green}) + \left|\frac{S_{Yellow}}{S}\right| E(S_{Yello}) + \left|\frac{S_{Red}}{S}\right| E(S_{Red})$

$= \frac{3}{16}\Big(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\Big) + \frac{7}{1}6\Big(-\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}\Big) + \frac{6}{16}\Big(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}\Big) = 0.9782$

$Information\_Gain(S, Color)$ = E(S) - 0.9782 = $1 - 0.9782 = 0.0218$ (Answer)

SELECT ATTRIBUTE : Type ; Values Type Takes = $< Normal, Water, Flying, Psychic >$

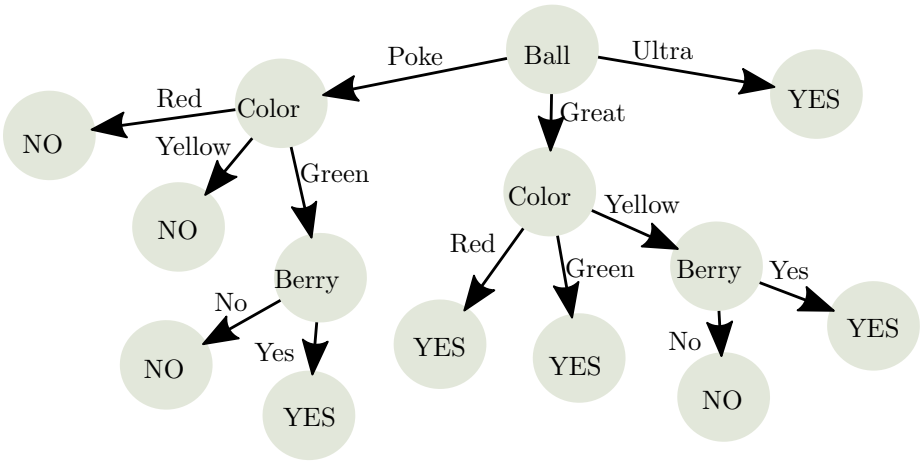
$\left|\frac{S_{Normal}}{S}\right| E(S_{Normal}) + \left|\frac{S_{Water}}{S}\right| E(S_{Water}) + \left|\frac{S_{Flying}}{S}\right| E(S_{Flying}) + \left|\frac{S_{Psychic}}{S}\right| E(S_{Psychic})$

$= \frac{6}{16}\Big(-\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}\Big) + \frac{4}{16}\Big(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\Big) + \frac{4}{16}\Big(-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\Big) + \frac{2}{16}\Big(-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\Big) = 0.8278$

$Information\_Gain(S, Type)$ = E(S) - 0.8278 = $1 - 0.8278 = 0.1722$ (Answer)

(d) Since Ball has the best Information Gain of 0.3252, Ball should be selected as the root of the decision tree as per ID3 (Answer)

(e) Below is a decision tree on the given dataset for pokemonGo, with the choice of attribute 'Ball' as the root.



(f) The prediction for the given test-set is shown in the below table:

Table 1: Berry vs Ball vs Color vs Type vs Prediction vs Actual-Caught

Berry	Ball	Color	Type	Prediction	Actual-Caught
Yes	Great	Yellow	Psychic	Yes	Yes
Yes	Poke	Green	Flying	Yes	No
No	Ultra	Red	Water	Yes	No

Thus, Accuracy $= \frac{1}{3} \times 100 = 33.33\%$ (Answer)

(g) The low accuracy of the small dataset suggests decision trees might be a bad choice of such a case.

Question-2:3: Gini Measure

Gini measure is defined as: $Gini(p_1, \ldots, p_k) = 1 - \sum_{i=1}^{k} p_i^2$

Gini(S) $= 1 - (p_{yes}^2 + p_{no}^2) = 1 - \left(\left(\frac{8}{16}\right)^2 + \left(\frac{8}{16}\right)^2\right) = 0.5$

SELECT ATTRIBUTE: Berry ; Values Berry takes $= < Yes, No >$

Information_Gain(S, Berry) = Gini(S) - $\sum_{v \in values(Berry)} \left|\frac{S_v}{S}\right| Gini(S) = \left|\frac{S_{yes}}{S}\right| Gini(S_{yes}) + \left|\frac{S_{no}}{S}\right| Gini(S_{no})$

$= 0.5 - \frac{7}{16}\left(1 - \left(\left(\frac{6}{7}\right)^2 + \left(\frac{1}{7}\right)^2\right)\right) + \frac{9}{16}\left(1 - \left(\left(\frac{2}{9}\right)^2 + \left(\frac{7}{9}\right)^2\right)\right) = 0.198$ (Answer)

SELECT ATTRIBUTE: BALL ; Values Ball can take $= < Poke, Great, Ultra >$

Information_Gain(S, Ball) $= 0.5 - \frac{6}{16}\left(1 - \left(\left(\frac{1}{6}\right)^2 + \left(\frac{5}{6}\right)^2\right)\right) + \frac{7}{16}\left(1 - \left(\left(\frac{4}{7}\right)^2 + \left(\frac{3}{7}\right)^2\right)\right) + \frac{3}{16}\left(1 - \left(\left(\frac{3}{3}\right)^2 + \left(\frac{0}{3}\right)^2\right)\right) = 0.1815$ (Answer)

SELECT ATRIBUTE: Color ; Values Color can take = $< Green, Yellow, Red >$

Information_Gain(S,Color) = $0.5 - \frac{3}{16}\left(1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right)\right) + \frac{7}{16}\left(1 - \left(\left(\frac{3}{7}\right)^2 + \left(\frac{4}{7}\right)^2\right)\right) + \frac{6}{16}\left(1 - \left(\left(\frac{3}{6}\right)^2 + \left(\frac{3}{6}\right)^2\right)\right) = 0.0149$ (Answer)

SELECT ATTRIBUTE: Type ; Values Type can take = $< Normal, Water, Flying, Psychic >$

Information_Gain(S, Type) = $0.5 - \frac{6}{16}\left(1 - \left(\left(\frac{3}{6}\right)^2 + \left(\frac{3}{6}\right)^2\right)\right) + \frac{4}{16}\left(1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right)\right) + \frac{4}{16}\left(1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right)\right) + \frac{2}{16}\left(1 - \left(\left(\frac{2}{2}\right)^2 + \left(\frac{0}{2}\right)^2\right)\right) = 0.09375$ (Answer)

(b) Using Gini measure, the attribute "Berry" should be the root of the decision tree. (Answer)
Since using Gini we get a different root node, therefore the trees derived using Gini and that derived using Entrophy will be different. However, in both the cases, the information gain of "Berry" and "Ball" were very close .

Question-2: Linear Classifier

---

(1) For the given dataset, the output of 1 corresponds to x4 being 1. and is -1 when all are 0. Thus for the all zero case, the bias term will be deciding the sign of the classifier. Hence we set the bias term to be -1. Since the sign function is defined as below,

sgn(f) = -1 if f < 0
sgn(f) = 1 if f ≥ 0

therefore, just ading x4 to the bias term brings the classifier to a positive value, resulting in +1 output. Hence the required linear classifier is:

Linear Classifier : $sgn(-1 + x_4)$ (Answer)

(2) For the given test set, following table lists down the prediction using the above classifier and comparison with the actual result

Table 2: x_1 vs x_2 vs x_3 vs x_4 vs prediction vs O

x_1	x_2	x_3	x_4	Prediction	O
1	0	1	1	1	1
0	1	0	1	1	1
1	0	1	0	-1	1
1	1	0	0	-1	1
1	1	1	1	1	1
1	1	1	0	-1	1
0	0	1	0	-1	-1

There are 3 mismatches in classifying the test data containing 7 samples.

Thus accuracy = $\frac{4}{7} \times 100 = 57.143\%$ (Answer)

(3) Given the rest of the missing data, the boolean expression of the function is $(x1 \vee x4)$. Hence, we need to adjust the classifier with the addition of x_1 in the sign function. Therefore the updated classifier is : Linear classifier : $sgn(-1 + x_4 + x_1)$

Question:3-SettingA: Experiments

1(a) The implemented decision tree using the ID3 algorithm is available in the Code folder of the tar file. The primary assumption made was the output label has two values. Three-valued labels for the output are not supported currently but will not be extremely difficult to tweak. Also, it uses Entrophy calculation as the parameter to measure the information gain for choosing the best attribute. It doesn't supports Gini measure yet. Besides these, there are no separate choices made other than what is discussed in the ID3 algorithm. The code is modular and python based.
1(b) The training accuracy of the decision tree on SettingA/training.data is 100%. The error of the decision tree on SettingA/training.data is 0% .
1(c) The error of the decision tree on SettingA/test.data is 0% . The test accuracy of the decision tree on SettingA/test.data is 100% .
1(d) Max-Depth of the decision tree while training on SettingA/training.data is 3 (excluding root node, including leaf node)

2(a) 6-fold Cross validation was run on the training data with depth list of [1,2,3,4,5,10,15,20]. However, the decision tree has a max-depth of 3, hence higher value of depth reports the same results. In the experiment we observe optimal choice is at depth of 2 as it reports the best average accuracy for the 6-fold cross validation. Infact, the accuracy does not changes for depth-2 and beyond, which means the most informative nodes has already been decided upon, and hence restricting the depth of the tree to 2 will restrict overfitting of the data. Below is a snapshot of the reported average accuracy and standard deviation for each depth.

```
<====================================================================>
*** Starting K-fold cross Validation with depth as Hyper ***

*** K-Fold cross Validation Ends ******************************

---------- K-Fold Experiment Result Detail ----------
At Depth Limit: 1     Average-Accuracy: 97.6452119309 %     Standard-Deviation: 5.12633590812
At Depth Limit: 2     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 3     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 4     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 5     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 10     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 15     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
At Depth Limit: 20     Average-Accuracy: 98.3254840398 %     Standard-Deviation: 3.74433151648
--------------------------------------------------------------------


Selected-Hyper-Depth =  2

##---------- Now train on the selected hyper-parameter ----------##
Final Training Accuracy with hyper-parameter Depth =  2 is  99.6860282575 %


------------ Testing with Test Data in current Setting -----------
Final Test Accuracy with hyper-parameter Depth =  2 is  99.7804610318 %
```

2(b) With the chosen depth with best accuracy , the depth of 2 is selected and the model is trained on the entire training set and then tested with the test data.
Accuracy of decision tree on SettingA/training.data = 99.686% (Answer)
Accuracy of decision tree on SettingA/test.data = 99.78% (Answer)

Table 3: K-fold cross Validation results for SettingA

Limit-Depth	Average-Accuracy(in %)	Standard-Deviation
1	97.645	5.126
2	98.325	3.744
3	98.325	3.744
4	98.325	3.744
5	98.325	3.744
10	98.325	3.744
15	98.325	3.744
20	98.325	3.744

Question:3-SettingB: Experiments

The root-node remains to be 'Odor' with this training set. The maximum depth of the tree is 9.
1(a) The accuracy of the decision tree on the SettingB/training data is 100%. Hence the error of the decision tree on the SettingB/training.data is 0.
1(b) The accuracy of the decision tree on the Setting/test.data is 92.97%. Hence, the error of the decision tree on the SettingB/test.data is 7.03%.
1(c) The accuracy of the decision tree with SettingA/training.data is 99.835%. Hence, the error of the decision tree on the SettingA/training.data is 0.165%.
1(d) The accuracy of the decision tree with SettingA/test.data is 99.947%. Hence, the error of the decision tree on the SettingA/training.data is 0.053%.
1(e) The maxDepth of the tree is 9.

2(a) 6-fold Cross validation was run on the split training data for SettingB with depth list of [1,2,3,4,5,10,15,20]. The tree has a max-depth of 9, however, we see the best average accuracy is achieved at around depth of 3, which is fixed as the hyperparameter to retrain the decision tree on the complete data set. Of the k training data sets, k iterations are made, selecting 1 of each as the test set in each iteration and the remaining as the training data. The root node evaluated everytime return 'Odor' as the best selection for root node. Below is a snapshot of the reported average accuracy and standard deviation for each depth. We see that on selecting a depth of 1, the accuracy falters to around 64%, since the tree is not decisive enough at single depth. However, by depth-3 it achieves good accuracy of 92.4% beyond which accuracy starts to reduce again. Thus, limiting our depth to depth of 3 seems the best choice as it will help not to overfit the decision tree on the training data and make quicker decisions with accuracy.

```
<=========================================================================>
*** Starting K-fold cross Validation with depth as Hyper ***

*** K-Fold cross Validation Ends ****************************

--------------- K-Fold Experiment Result Detail ---------------------
At Depth Limit: 1     Average-Accuracy: 64.652014652 %     Standard-Deviation: 34.2921513321
At Depth Limit: 2     Average-Accuracy: 90.3453689168 %     Standard-Deviation: 9.73470640907
At Depth Limit: 3     Average-Accuracy: 92.4385138671 %     Standard-Deviation: 4.29294091598
At Depth Limit: 4     Average-Accuracy: 92.2815279958 %     Standard-Deviation: 3.21395318563
At Depth Limit: 5     Average-Accuracy: 92.0722135008 %     Standard-Deviation: 2.98238726703
At Depth Limit: 10     Average-Accuracy: 91.7582417582 %     Standard-Deviation: 2.98514045226
At Depth Limit: 15     Average-Accuracy: 91.7582417582 %     Standard-Deviation: 2.98514045226
At Depth Limit: 20     Average-Accuracy: 91.7582417582 %     Standard-Deviation: 2.98514045226
---------------------------------------------------------------------


Selected-Hyper-Depth =  3

##---------- Now train on the selected hyper-parameter ---------------##
Final Training Accuracy with hyper-parameter Depth =  3 is  95.9968602826 %


------------ Testing with Test Data in current Setting -----------
Final Test Accuracy with hyper-parameter Depth =  3 is  93.6333699232 %
```

Table 4: K-Fold cross validation for setting B

Limit-Depth	Average-Accuracy(in %)	Standard Deviation
1	64.652	34.292
2	90.345	9.734
3	92.438	4.292
4	92.281	3.213
5	92.072	2.982
10	91.758	2.985
15	91.758	2.985
20	91.758	2.985

Hence, the selected hyperparameter value of depth = 3
So, depth of 3 should be chosen as the best, since it gives the best average accuracy and allows us to not overfit the training data such that noise in the training data can be eliminated.

2(b) On retraining the decision tree with the selected hyper-parameter value and checking accuracy of its predictions on the training data and test data of SettingB, the followingh results are seen
Training accuracy on SettingB/training.data = 95.996%
Test accuracy on SettingB/test.data = 93.633

Question-3:SettingC: Experiments

(1) The decision tree is updated here to include the feature of working with missing attribute values. We use three different methods to determine the which method fits bests to train the decision tree. The program requires a '-mf' switch to enable method evaluation. The three methods and the assumptions made are briefly described below:

METHOD1: In this method, the missing feature is set as the majotiry feature value for that attribute. The whole training data(the split ones for k-fold) are first accumulated together, and the missing feature value is determined. Then they are resplit for k-fold cross-validation(CV).

METHOD2: In this method, the missing feature is set to the majority value of that label. Like method-I, we accumulate the entire training data and determine the missing feature values by iteratively traversing the example set and finding the majority value of that label corresponding to the missing feature

.

METHOD3: In this method , the missing feature is treated as a special feature. We update the GlobalAttribute datastructure used to tract the linking of features with their values such that the feature with a missing value gets an additional 'special value' for traiining.
During the test phase with the actual test dataset, for method2, we do not look into the test set since avert the affect of the outputs in the test set inducing any bias in the association of the missing feature value. For both Method-1 and Method-2, the test set is cleaned as in Method1, that is, by the majority feature value. For Method3, test set clean up is not required since the missing feature value is treated as a special feature value.

(2) 6-fold cross validation eas performed using all the above three methods. Below is a snapshot of the reported accuracy and standard deviation for each method.

```
<===========================================================================>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~ Starting Training Method-I ~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*** Starting K-fold cross validation for missing feature/Method-1 ***

Method-I =  98.6769570011 %

*** Method-1 K-fold CV ends here ************************************

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~ Starting Training Method-II ~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*** Starting K-fold cross validation for missing feature/Method-2 ***

Method-II =  98.6769570011 %

*** Method-3 K-fold CV ends here ************************************

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~ Starting Training Method-III ~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*** Starting K-fold cross validation for missing feature/Method-3 ***

Method-III =  100.0 %

*** Method-3 K-fold CV ends here ************************************

--------------- Method-1/2/3 k-Fold Experiment Result Detail ----------------------
Method-2           Average-Accuracy: 98.6769570011 %              Standard-Deviation: 2.95841408269
Method-3           Average-Accuracy: 100.0 %              Standard-Deviation: 0.0
Method-1           Average-Accuracy: 98.6769570011 %              Standard-Deviation: 2.95841408269


Method3 Won - Train Full data using Method3 and Test
TrainingAccuracy in Method3 =  100.0 %

TestAccuracy in Method3 =  100.0 %
```

Table 5: Method-1/2/3 k-fold Experiment on Setting C

Method	Average-Accuracy(in %)	Standard-Deviation
Method-1	98.676	2.958
Method-2	98.676	2.958
Method-3	100	0

As evident from the above figure, Method-1 and Method-2 reports an accuracy of 98.67% with a standard deviation of 2.958 while Method-3 reports an accuracy of 100% with standard deviation of 0. Thus MEthod3 wins the race.

2(3) From the 6-fold experiment, Method-3, was the winner. Next, we retrain the decision tree using the Method-3 on the entire training dataset of SettingC and test it on the SettingC/test.data , treating the missing feature value as 'special'.
Training Accuracy in Method3 = 100%
Test Accuracy in Method-3 = 100%

Thats all for now. Check out the code